COOLEY LLP
BOBBY GHAJAR (198719)
(bghajar@cooley.com)
TERESA MICHAUD (296329)
(tmichaud@cooley.com)
COLETTE GHAZARIAN (322235)
(cghazarian@cooley.com)
1333 2nd Street, Suite 400
Santa Monica, California 90401
Telephone: (310) 883-6400

MARK WEINSTEIN (193043)
(mweinstein@cooley.com)
KATHLEEN HARTNETT (314267)
(khartnett@cooley.com)
JUDD LAUTER (290945)
(jlauter@cooley.com)
ELIZABETH L. STAMESHKIN (260865)
(lstameshkin@cooley.com)
3175 Hanover Street
Palo Alto, CA 94304-1130
Telephone: (650) 843-5000

CLEARY GOTTLIEB STEEN & HAMILTON LLP
ANGELA L. DUNNING (212047)
(adunning@cgsh.com)
1841 Page Mill Road, Suite 250
Palo Alto, CA 94304
Telephone: (650) 815-4131

[Full Listing on Signature Page]

Counsel for Defendant Meta Platforms, Inc.

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

RICHARD KADREY, *et al.*,

Individual and Representative Plaintiffs,

v.

META PLATFORMS, INC., a Delaware corporation;

Defendant.

Case No. 3:23-cv-03417-VC-TSH

DECLARATION OF PROFESSOR LYLE UNGAR, PH.D. IN SUPPORT OF META'S MOTION FOR SUMMARY JUDGMENT

I, Lyle Ungar, Ph.D., declare:

1. I am over the age of 18 and am competent to make this declaration. I have been engaged by Meta Platforms, Inc. ("Meta") as a technical expert to provide my opinion regarding certain aspects of the development, training, and operation of large language models ("LLMs"), in particular the Meta Llama LLMs. I submitted an opening expert report in this case on January 10, 2025 and a rebuttal report on February 3, 2025, and was deposed in this matter on February 26, 2025. The opinions below provide a concise subset of the opinions as set forth in my two expert reports. For convenience and ease of reference, I have included footnotes in each paragraph or section indicating where the text appears in my Opening and/or Rebuttal Reports. The statements in my Opening and Rebuttal Reports are true and accurate to the best of my knowledge, information, and belief. Except as otherwise stated herein, I make this declaration based on my personal knowledge and professional expertise, as well as Meta documents (as disclosed in my reports) and source code produced in this action as well as research and experiments conducted by me or under my direction and supervision.

Professional Background and Qualifications

2. I am a Professor of Computer and Information Science at the University of Pennsylvania, where I have been a faculty member since 1984. I serve as a faculty member in the Graduate Groups of Bioengineering, Genomics and Computational Biology (in the School of Medicine), Operations Information and Decisions (in the Wharton School), and Psychology (in the School of Arts and Sciences). I am also a Distinguished Research Fellow at the Annenberg Public Policy Center at the University of Pennsylvania and a member of the Center for Cognitive Neuroscience (CCN), also at the University of Pennsylvania. I received a PhD from the Massachusetts Institute of Technology in Chemical Engineering in 1984 and a BS from Stanford University in Chemical Engineering (with distinction). I previously served as the Graduate Chair for the Computer and Information Science (CIS) department, Associate Director of the Penn Center

for BioInformatics (PCBI), and on the Executive Committee for the Genomics and Computational Biology (GCB) group, all at the University of Pennsylvania.[1]

3. In the research community, I served as chair for the IEEE International Conference on Bioinformatics and Biomedicine, the ACM International Conference on Knowledge Discovery and Data Mining, served as Associate Editor for the Journal of Machine Learning Research, and served on the committees for the American Association for Artificial Intelligence (AAAI), the ACM Knowledge Discovery and Data Mining (KDD), and the IEEE International Conference on Data Mining (ICDM), among others.[2]

4. I was named the Leshner Fellow by the American Association for the Advancement of Science (AAAS) for the 2020-2021 Artificial Intelligence cohort. The Leshner Leadership Institute seeks to empower scientists to lead high-impact public engagement and advocate for institutional change to further support public engagement by scientists.[3]

5. I publish extensively in leading journals for machine learning, artificial intelligence, data science, statistical analysis, medicine, and psychology. Per Google Scholar, I have more than 48,000 citations; an h-index of 99, which means 99 of my publications have been cited over 99 times; and an i-10 index of 360, which means 360 of my publications have been cited over 10 times. I have hundreds of published articles and hold several granted patents.[4]

6. My core expertise lies in machine learning, deep learning, and natural language processing. This includes machine learning and data mining techniques involving the analysis of LLMs, causal and interpretable models, clustering and collaborative filtering, feature selection, and statistical relational learning. I also specialize in natural language processing methods underlying chatbots, analyzing natural language processing generated conversations, and text mining user-generated content.[5]

---

[1] Ungar Opening Report, ¶¶ 6-7.

[2] Ungar Opening Report, ¶ 8.

[3] Ungar Opening Report, ¶ 9.

[4] Ungar Opening Report, ¶ 10. Note I updated the citation count and i-10 index values in the text, as they have increased since service of my opening report in January 2025.

[5] Ungar Opening Report, ¶ 11.

7. As a Professor, I teach undergraduate and graduate classes covering topics including Artificial Intelligence, Machine Learning, Data Mining, Model Building with Modern Statistics, and Big Data Analytics, among others. Specific classes I teach in this area, for example, include "Advanced Topics in Deep Learning," which covers advanced LLM topics like attention architecture and retrieval-augmented generation, and "Conversations and Conversational Bots," which covers statistical learning theory, among others. I've also supervised more than 30 master's and doctoral students, including working on natural language processing techniques and with models such as OpenAI's GPT, Google's Gemma, Anthropic's Claude, and Meta's Llama.[6]

Overview of AI and Neural Networks

8. Large language models (LLMs) have rapidly gained popularity across varied domains in the past few years as interaction with these models has become increasingly accessible and useful. The underlying technology powering LLMs, however, is built on decades of innovation and developments in computer science and the field of natural language processing (NLP).[7]

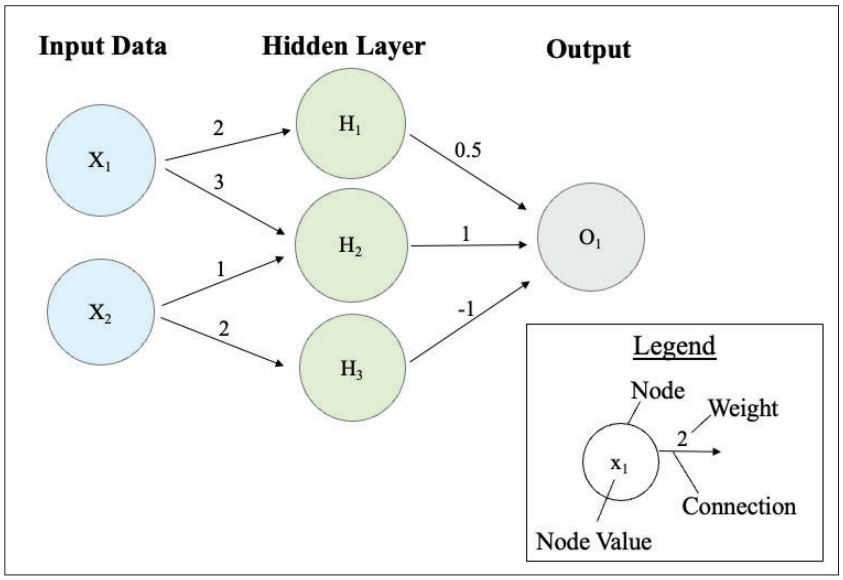
9. Many modern technological tasks are performed by computers that are programmed to follow sets of instructions, known as algorithms, which often comprise multiple steps that a computer is instructed to follow to achieve desired results. Historically, programmers created algorithms that required explicitly defining every possible step, scenario, and rule necessary, known as hard coding the instructions. Artificial intelligence, by contrast, involves methods and techniques (including algorithms) that enable computers to simulate human intelligence by capturing, understanding, and learning from complex patterns and relationships in data to solve problems and achieve a desired result, without requiring explicit hard coding from programmers to account for all possible scenarios.[8]

10. Dating back to the 1950s, early AI systems were programmed to be rule- and logic-based, and performed tasks that were once considered to require human knowledge, like playing checkers or chess. These systems made decisions and performed actions based largely on

[6] Ungar Opening Report, ¶ 13.
[7] Ungar Opening Report, ¶ 16.
[8] Ungar Opening Report, ¶¶ 24-25.

predefined logical rules often compiled by human beings, similar in function to a list of "IF-THEN-ELSE" rules commonly used in computer programming. By the 1970s, computer scientists had developed AI systems commonly known as "expert systems" that were designed to simulate the decision-making of a human being with expertise in particular fields by collecting a large number of predefined rules in particular fields to make decisions and perform actions within those fields. But because these earlier AI systems had the drawback of having to rely on explicitly defined knowledge and rules and Boolean logic, they were limited in their ability to generalize and address situations that fell outside their predefined ruleset.[9]

11. In parallel, and picking up in the 1980s, researchers were investigating alternative techniques commonly referred to as ***neural networks***, which took a radically different approach from AI expert systems. Neural networks, as I will explain below, are a foundation of modern LLMs. The basic idea behind neural networks—and the motivation for their name—was to try to design a computing system that emulated the functions of the human brain. Figure 1 below shows a diagram showing of a highly simplified neural network, to illustrate some of the key concepts:



As pictured, neural networks are composed of a series of connected nodes (shown as the circular structures above), which are also referred to as neurons. In a neural network, the input data is preprocessed and converted into numbers that are readable by a computer, then passed through a

[9] Ungar Opening Report, ¶ 25.

series of interconnected network layers of the neural network. The first layer (on the left) is called the ***input*** layer, the final layer (on the right) is called the ***output*** layer, and interior layers are known as ***hidden*** layers. The input layer serves as the starting point for the neural network's processing. The hidden layers represent the neural network's internal calculations, applying a series of mathematical operations to extract features and capture sophisticated patterns across the entirety of the input data. The output layer represents the network's prediction about the input.[10]

12. The lines connecting the nodes are commonly referred to as ***connections***, and each has a numerical value commonly called a ***weight***. The neural network's weights are the fundamental building block of the network, as they determine how important a corresponding aspect of the input data is to the next layer of nodes and control the predictions that the model generates. For example, the exemplary connection between the node x_1 in the input layer above and the top node H_1 in the hidden layer has a value of "**2**," which indicates that that hidden layer node will receive the input value of X_1, multiplied by 2. With respect to the next node in the hidden layer, H_2, it has two connections—a connection of "**3**" from x_1 and "**1**" from X_2. That node, therefore, will receive the sum of (1) the input value of X_1 multiplied by 3, and (2) the input value of X_2 multiplied by 2. In other words, each node in a hidden layer calculates a value that represents the weighted sum of the outputs from the previous layer. This process propagates throughout the layers in the neural network.[11]

13. Although the simplified diagram above shows only one hidden layer and six connections, modern LLMs are dramatically larger with dozens of layers and billions of connections. For example, the largest version of Meta's Llama 3 currently includes 405 billion tunable weights distributed across 126 layers, with each weight represented by a decimal value. A sample of a neural network's weights (showing only 100 weights out of billions total) is shown below:

[10] Ungar Opening Report, ¶¶ 26-27.

[11] Ungar Opening Report, ¶ 28.

```
[ 1.2516975e-06, -1.7881393e-06, -4.3511391e-06,  8.0466270e-06,
  1.9073486e-06, -5.6028366e-06,  3.0994415e-06,  1.1920929e-06,
 -6.7949295e-06, -1.6689301e-06, -4.4703484e-06, -4.4107437e-06,
 -7.1525574e-07, -8.4638596e-06,  2.1457672e-06,  1.0251999e-05,
 -4.7683716e-07, -1.5497208e-06,  1.6093254e-06,  1.1324883e-06,
  2.8610229e-06,  9.4771385e-06,  3.3378601e-06, -2.8014183e-06,
 -1.2874603e-05, -2.8014183e-06,  5.6028366e-06, -1.1324883e-06,
 -3.3378601e-06, -2.9802322e-06, -2.3841858e-07,  1.4305115e-06,
  9.1791153e-06,  2.5629997e-06,  1.9669533e-06,  9.5367432e-07,
 -1.1086464e-05, -5.7220459e-06,  3.9339066e-06, -1.1026859e-05,
  7.2121620e-06,  1.8477440e-06,  4.5895576e-06,  2.2053719e-06,
  1.3113022e-06, -2.8610229e-06, -1.4841557e-05, -6.4373016e-06,
  2.6226044e-06,  6.6757202e-06, -2.6226044e-06,  8.3446503e-06,
  1.6093254e-06, -1.3053417e-05,  4.6491623e-06,  7.8082085e-06,
 -5.6028366e-06, -6.3180923e-06, -3.5762787e-07,  9.2387199e-06,
  5.3644180e-06, -3.9339066e-06, -2.4437904e-06, -4.6491623e-06,
 -6.9737434e-06,  1.9073486e-06,  4.0531158e-06, -2.9206276e-06,
  2.6822090e-06,  9.1791153e-06, -6.1988831e-06,  5.6624413e-06,
  1.4901161e-06,  8.1658363e-06, -3.6954880e-06,  7.0333481e-06,
 -1.6689301e-06,  2.8610229e-06,  8.4042549e-06, -5.1259995e-06,
  8.2254410e-06, -3.6954880e-06,  9.5367432e-06,  1.5497208e-06,
 -4.2915344e-06,  2.0265579e-06, -2.0265579e-06, -1.5497208e-06,
 -8.9406967e-06, -1.1920929e-07,  4.7683716e-06,  2.6226044e-06,
  6.9737434e-06,  9.5963478e-06,  3.4570694e-06, -7.6889992e-06,
-1.9073486e-06, -5.0067902e-06,  6.4373016e-06,  5.3644180e-06, ... ]
```
[12]

14. During the ***training*** of a neural network, as I will explain in more detail below, input data is provided to the network and used to adjust the weights through a series of training algorithms, which shape the neural network's output predictions to more closely match expected output values. Input data is not stored in the model after training is concluded, and a neural network's weights are not a direct representation of the input data, as they encode generalized features learned from the training dataset such as syntactic, semantic, and contextual relationships across the entire dataset. One of the key benefits of neural networks is that they do ***not*** require that a human being come up with predefined or preprogrammed logical rules. Their capabilities instead derive from the structure of the network and the connections/weights, which through training, enable the network to recognize patterns and relationships in the data and make predictions. This allows neural networks to generalize and make predictions based on new input.[13]

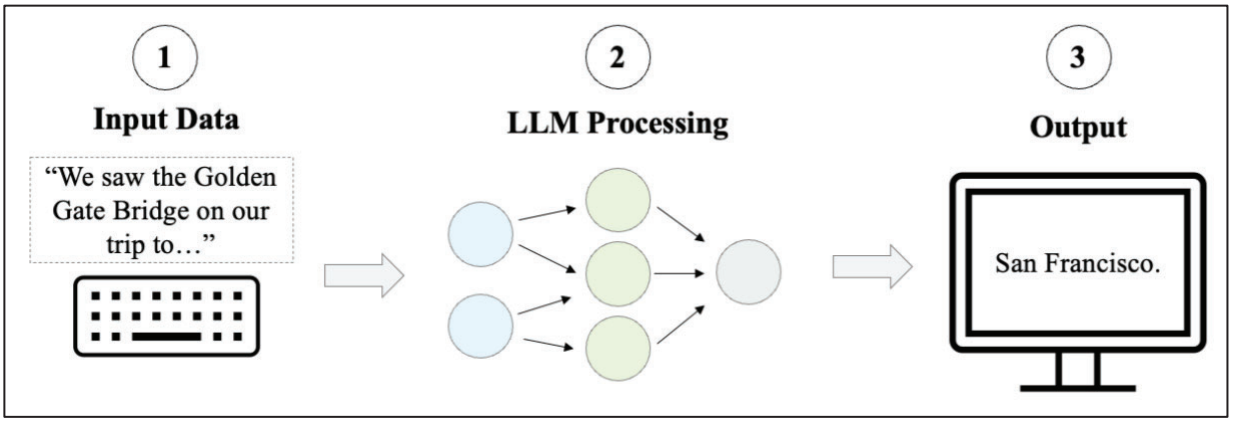
[12] Ungar Opening Report, ¶ 49. The exemplary numbers in the table show exponential notation of multiplication by a power of 10. For example, the first weight on the top-left ("1.2516975e-06"), therefore, represents the value of 0.0000012516975.

[13] Ungar Opening Report, ¶¶ 29-30.

Overview of LLMs

15. A ***Large Language Model (LLM)*** is a type of deep learning model that is trained on vast amounts of text data, designed to understand the patterns and structures of natural language, and generate language as output when prompted. Notable examples of LLMs include Meta's Llama, OpenAI's GPT-4, and Anthropic's Claude. LLMs fall under ***Generative Artificial Intelligence (GenAI)*** models due to their capability of *generating* language as output. With advancements in natural language processing research, LLMs perform well across a wide range of applications, such as question answering, text classification, text summarization, content creation, and coding, among others.[14]

16. LLMs generate text in response to ***prompts***, which are passages of text that are input to the LLM. Prompts can be any sentence, question, or request, and the LLM's goal is to produce a sensible response to the prompt, referred to as the LLM's ***output***. The figure below provides an example showing an LLM with a prompt ("We saw the Golden Gate Bridge on our trip to…") and the LLM produces an output ("San Francisco").[15]



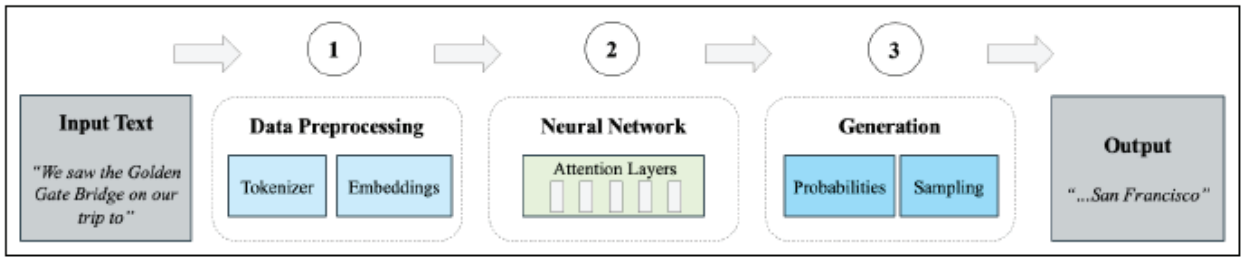
17. In the figure above, a user inputs a *prompt* (1), which is processed by the LLM (2) to produce an *output* (3) that logically continues the prompt. LLMs are neural networks that typically use a transformer architecture to predict the next word in a sequence.

18. Broadly speaking as discussed below, LLMs produce outputs from prompts in three steps. First, inputs are turned from text (***prompts***) into numbers (***embeddings***). Then, these

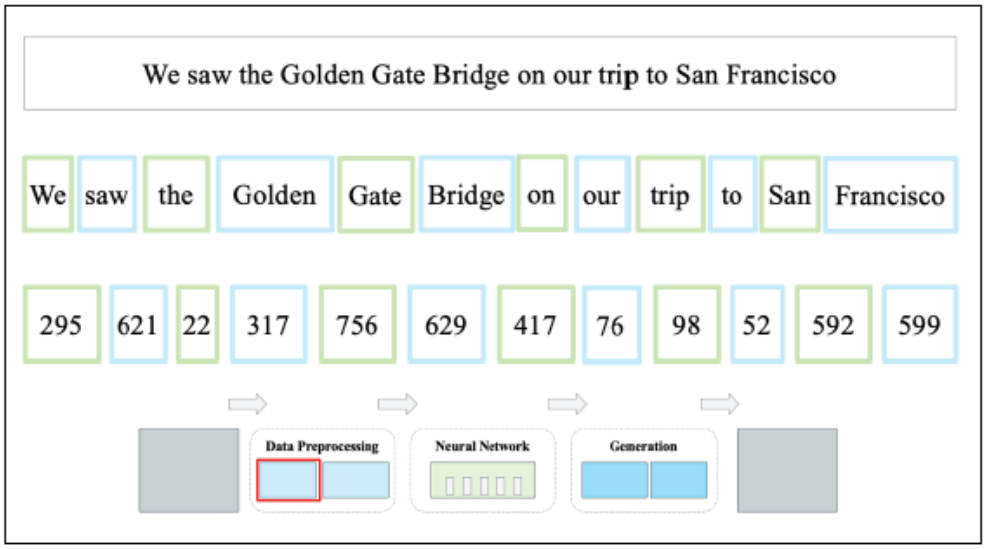
[14] Ungar Opening Report, ¶ 63.

[15] Ungar Opening Report, ¶ 64.

numerical inputs pass through a very large neural network, in which billions of weights are iteratively adjusted to produce a list of ***probabilities*** corresponding to the next word prediction for all words in the vocabulary. Finally, the model chooses an output word based on these probabilities, providing the model's completion of the prompt. The figure shows this process, which is explained in the following steps.[16]



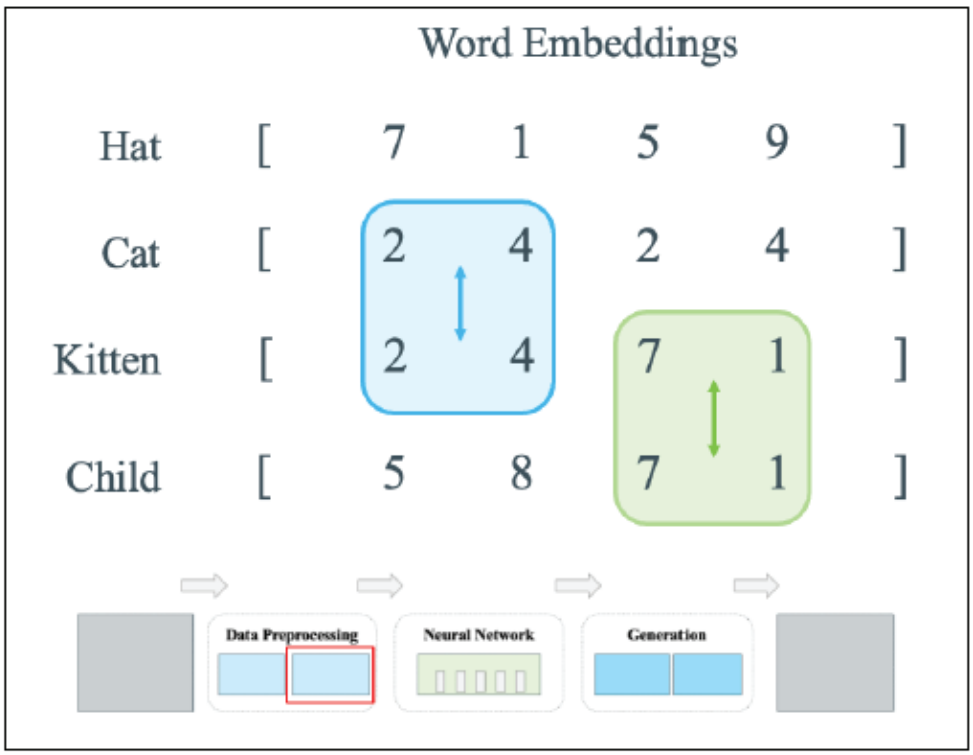
19. Because neural networks, including LLMs, can operate only on numbers—all non-numeric data such as text, images, and audio must first be broken down into smaller units and then converted into numbers to be processed by a neural network. This process is collectively known as data preprocessing. In LLMs, preprocessing has two steps: first, text is ***tokenized***, i.e., broken into word pieces (as there are too many words to remember them all) where each token is assigned a number. Then it is ***embedded***, and further transformed into numerical representations of the meaning of the text. The figure below shows an example of tokenization.[17]



[16] Ungar Opening Report, ¶ 67.

[17] Ungar Opening Report, ¶¶ 68-69.

20. While tokenized representations of words are numerical and computer-readable, they are not very informative in terms of understanding the meaning of underlying words as they do not capture similarities, differences, or relationships between words. As one example, the Llama 2 tokenized representations of "cat" (7159) and "kitten" (2424) have no connection and reveal no relationship between the words. Because it is desirable to have informative numerical representations of words that capture relationships between similar words numerically, LLMs utilize ***word embeddings***, which are many-dimensional ***vectors*** that encode words' meanings. The figure below provides examples of four-dimensional word embeddings.[18]

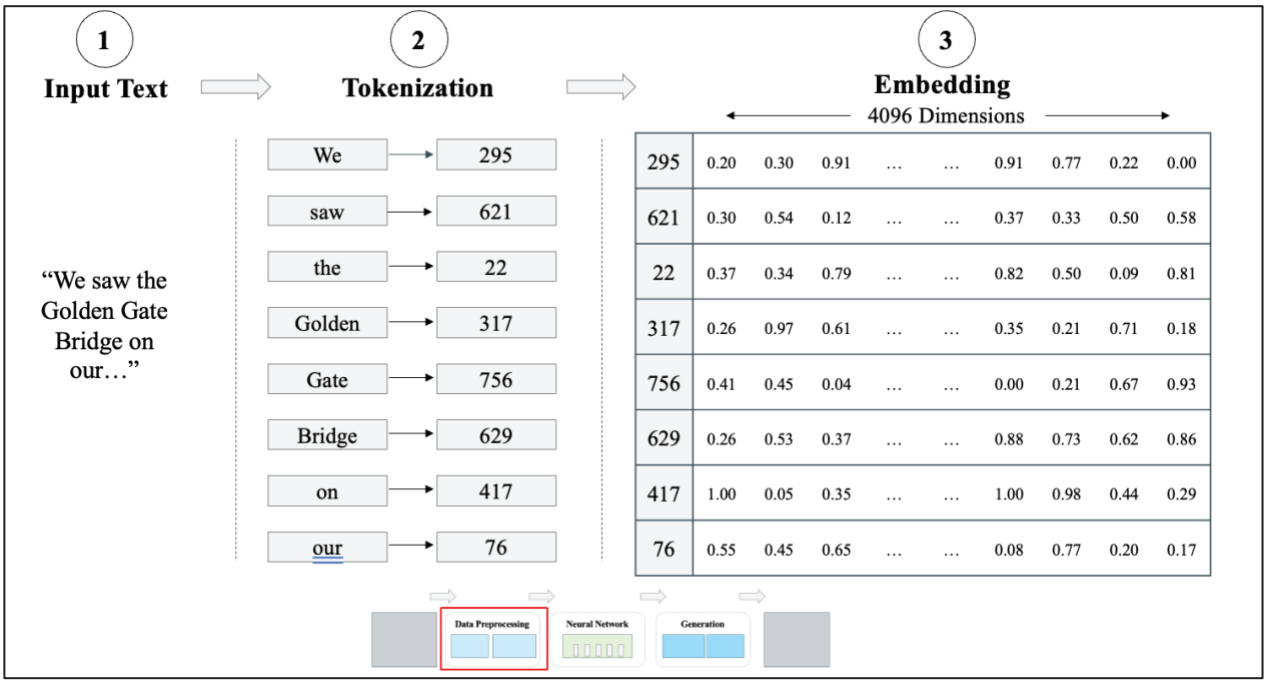


21. In the above example, despite their phonetic similarity, "Hat" and "Cat" have different meanings and thus very different embeddings. In contrast, "Cat" and "Kitten" are different phonetically but have similar meanings and are thus similar along some dimensions. "Kitten" and "Child" are also phonetically different but similar in meaning, thus having similarity along other dimensions, as their similarity is distinct from "Cat" and "Kitten." In this example, the two leftmost dimensions can be understood to convey the species of the words, while the two

[18] Ungar Opening Report, ¶ 73.

rightmost dimensions capture the age of the words.[19] In practice, LLM word embeddings have hundreds or thousands of dimensions, rather than just the four shown above. [20]

22. The full preprocessing process, using specifications from Llama 2 7B as exemplary, is illustrated below, from input text to word embeddings. Once input tokens are transformed into their respective embeddings, those embeddings are input into a neural network to produce predictions for subsequent tokens.[21]

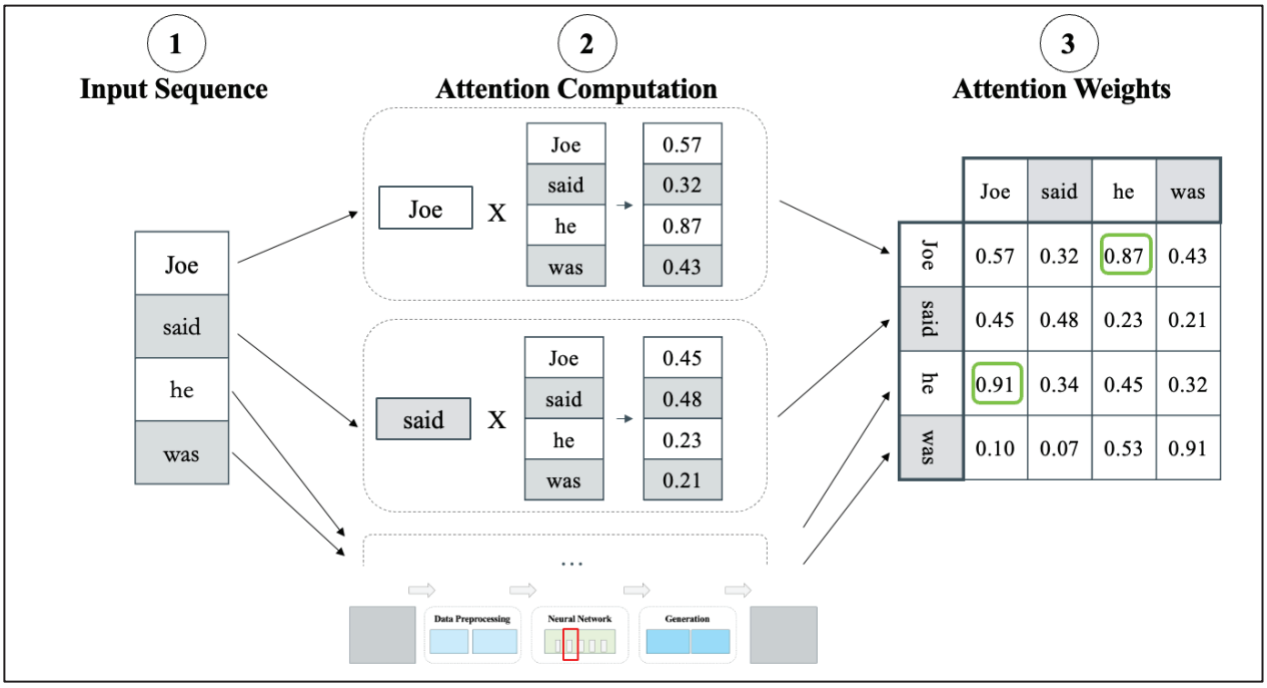


23. While an LLM's word embeddings capture many aspects of a word's meaning, they do not provide any information about the context in which a word appears. For example, in the phrases "he tied the ribbon into a bow" and "he fired an arrow from his bow," the word "bow" would correspond to the same embedding in both cases after preprocessing. LLMs address this issue via "attention," a mathematical computation that compares each object (token) in one sequence (sentence/paragraph) with the objects in another sequence.[22]

24. In particular, LLMs use "self-attention," which compares each token in a text with every other token in the same text. It allows the LLM to model relationships between words in a

[19] Ungar Opening Report, ¶ 73.
[20] Ungar Opening Report, ¶ 74.
[21] Ungar Opening Report, ¶ 74.
[22] Ungar Opening Report, ¶¶75, 77.

sentence or paragraph, thereby placing the words in the context of the sentence or paragraph in which they belong. The figure below shows an example of a trained attention layer that computes relationships between pronouns and their corresponding nouns. Each word in the sequence "Joe said he was" is compared with every other word (including itself) to produce a table of attention weights. In this example, the attention model predicts noun-pronoun relationships, so "Joe" and "he" have high weights between them, while connections between other words have smaller weights. This is only one example of many potential relationships an attention layer could detect:[23]

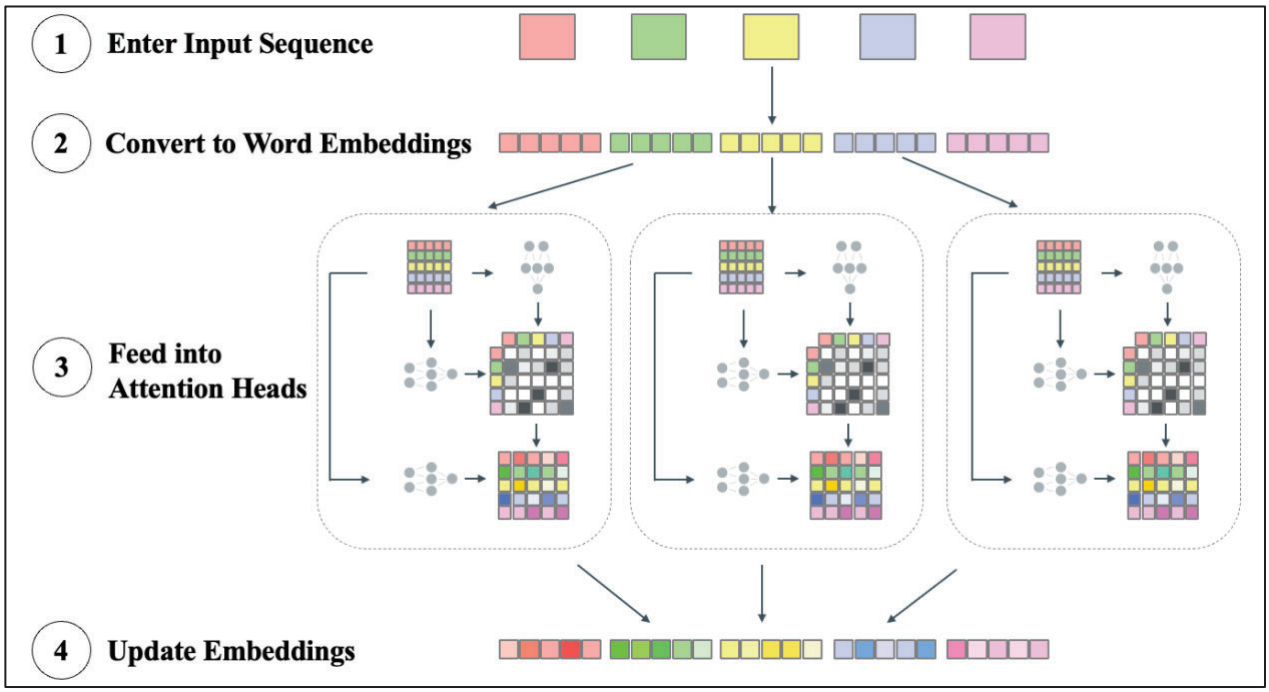


25. A single relationship between words, like the one displayed above, is useful but insufficient to capture all relationships between tokens in the input sequence. To understand different relationships between all tokens, LLMs compute attention many times in each layer of their neural networks. Each attention computation is conducted within an ***attention head*** and approximates a different relationship between tokens. For example, one head may relate words in sentences with their associated verbs, and another may connect words in sentences with that sentence's closing punctuation. This architecture, called ***multi-headed attention***, allows the LLM to capture many useful relationships between tokens in an input sequence.[24]

[23] Ungar Opening Report, ¶ 77.

[24] Ungar Opening Report, ¶ 78.

26. The figure below shows a representation of a single layer of multi-headed attention in an LLM. Modern LLMs tend to use many attention heads. For example, the smallest version of Llama 2 features 32 attention heads per layer, while the largest has 64 heads per layer. After each attention head finishes its computations, the outputs from all heads are combined, creating a transformed version of the original token embeddings, such that it contains additional information about tokens and their relationships. The multiheaded attention process constitutes a single layer of an LLM's neural network.[25]

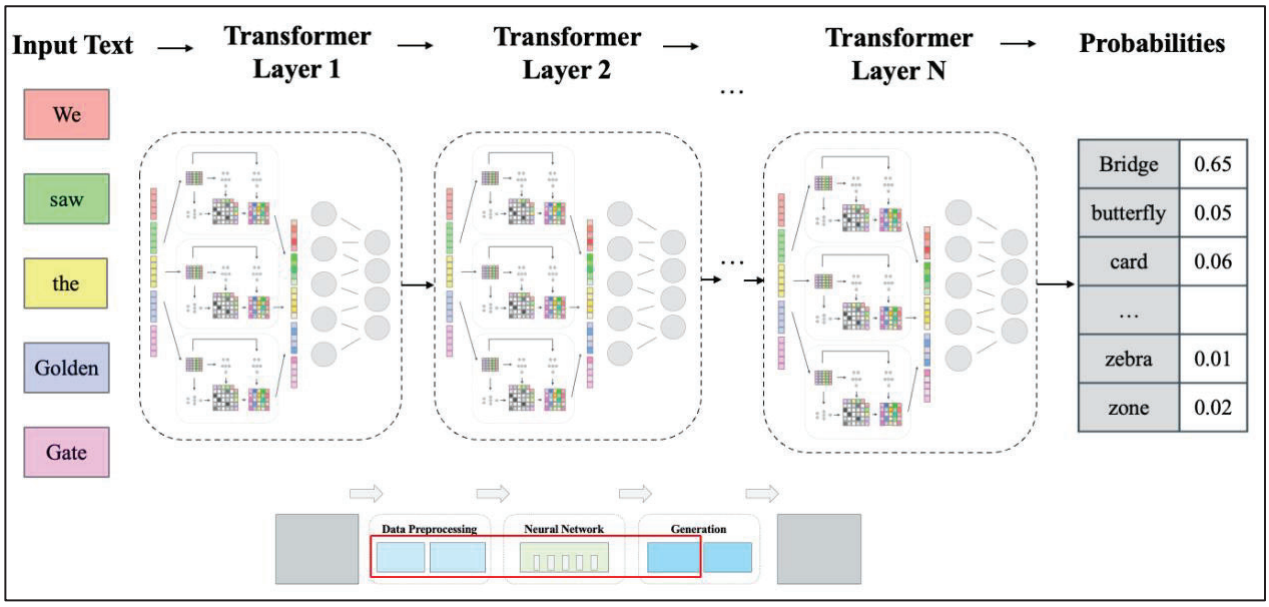


27. Each LLM neural network layer outputs vectors that are the same dimension as an LLM's word embeddings. The outputs from one layer are passed into subsequent LLM layers, where attention is computed on those outputs, which are then passed to the next layer, and so on. Each layer modifies embeddings to better capture the contextualized meaning of each word in the input text. This type of neural network, featuring stacked attention layers, is a widely used architecture called a transformer. Transformers are the underlying architecture in modern LLMs, and many other neural networks that operate on text, images, audio, and other data. [26]

[25] Ungar Opening Report, ¶ 78.

[26] Ungar Opening Report, ¶ 79.

28. Rather than outputting transformed embeddings or predicted tokens directly, the final layer of a transformer outputs a probability for each token from the model's tokenizer vocabulary. Each probability represents the model's prediction with respect to a corresponding token as being the next token for the input sequence. For example, in the illustration below, the LLM has estimated there is a 65% chance that the next token in the sequence is "Bridge," a 5% chance that the next token is "butterfly," and so on.[27]



29. To generate texts longer than a single word, LLMs generate text autoregressively, meaning that the output of one pass through the model is fed back into the model to generate the next tokens in the sequence. For example, if "We saw the Golden Gate" was input into an LLM, and "Bridge" was the LLM's predicted next token, then "We saw the Golden Gate Bridge" would be input into the LLM to obtain the next predicted word in the sequence. This allows LLMs to generate text indefinitely by appending their output predictions to input text and applying their transformer architecture to the newly completed sequence. Each new token requires the transformer model to newly compute multi-headed attention at each layer.[28]

[27] Ungar Opening Report, ¶ 80.

[28] Ungar Opening Report, ¶ 83.

Overview of LLM Training

30. Broadly, there are two phases to training an LLM: ***pretraining*** and ***post-training***. I understand that the Plaintiffs in this case have alleged a claim of copyright infringement based on allegations that their works were used by Meta in the ***pretraining*** of the Llama models, and as such, I will focus on that aspect of training in my discussion below.[29]

31. LLMs are trained by utilizing a complex set of statistical processes in which billions of weights are iteratively updated based on probability distributions. "Pretraining" is the initial phase where the LLM is trained on text data. During this phase, the model is trained to predict the next word in a sequence, enabling it to learn general patterns in language and knowledge. The goal of pretraining is for the model to generalize across many types of tasks, contexts, and language structures by training on many distinct examples of language.[30]

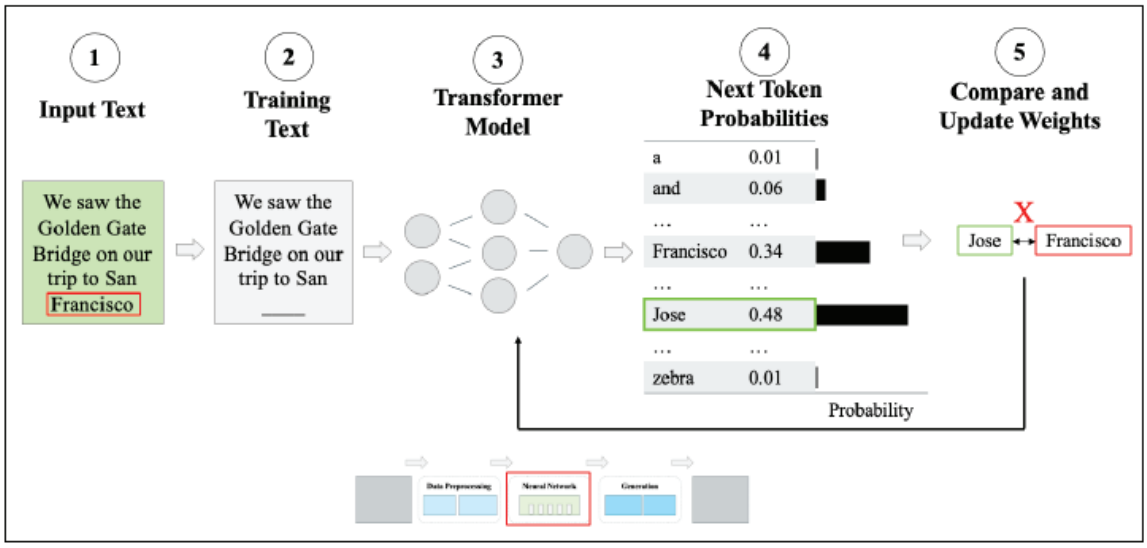
32. In the pretraining process, LLMs are trained across billions of steps via a complex, computation-intensive process to predict the next token in a sequence. At each step, practitioners obtain a sequence of text from the LLM's training data, then remove a subsequent token from the sequence. The LLM then predicts the hidden token. While the neural network makes its prediction, each weight in the network is tracked to compute its effect on the final prediction. The LLM's prediction is compared to the actual deleted token, with two possible outcomes: if the LLM's prediction was correct, the weights are adjusted to reinforce the accurate prediction, and if the prediction was incorrect, the weights are updated to minimize error and improve future prediction. Every weight that contributes to a prediction is updated at each step, including the model's word embeddings which begin as random numbers and are gradually improved throughout training. The training step is repeated billions, even trillions of times (based on the size of the training dataset)

---

[29] Post-training generally refers to the process by which a pretrained LLM is refined to align it with specific tasks, domains, or performance goals. Unlike pretraining, which builds foundational knowledge, post-training adapts a pretrained model to a specific task or domain by using smaller, more targeted datasets. For example, "supervised finetuning" is a method by which a pretrained LLM is fine-tuned using specialized task-specific data where both the inputs and the desired outputs are explicitly provided to the model in the form of simulated dialogues with example prompts and model responses. Another finetuning technique, reinforcement learning from human feedback, is an iterative process involving direct human feedback to guide model outputs toward developer preferences. Ungar Opening Report, ¶¶ 102-103.

[30] Ungar Opening Report, ¶ 83.

until the LLM has made predictions about all of the trillions of words in its training dataset. Along the way, the weights of the LLM are adjusted to make more accurate predictions about hidden words by gaining (implicit) information about words, language grammar, and facts across the entirety of the training set.[31] To concretize the training process, the figure below illustrates an example of the LLM training process.[32]



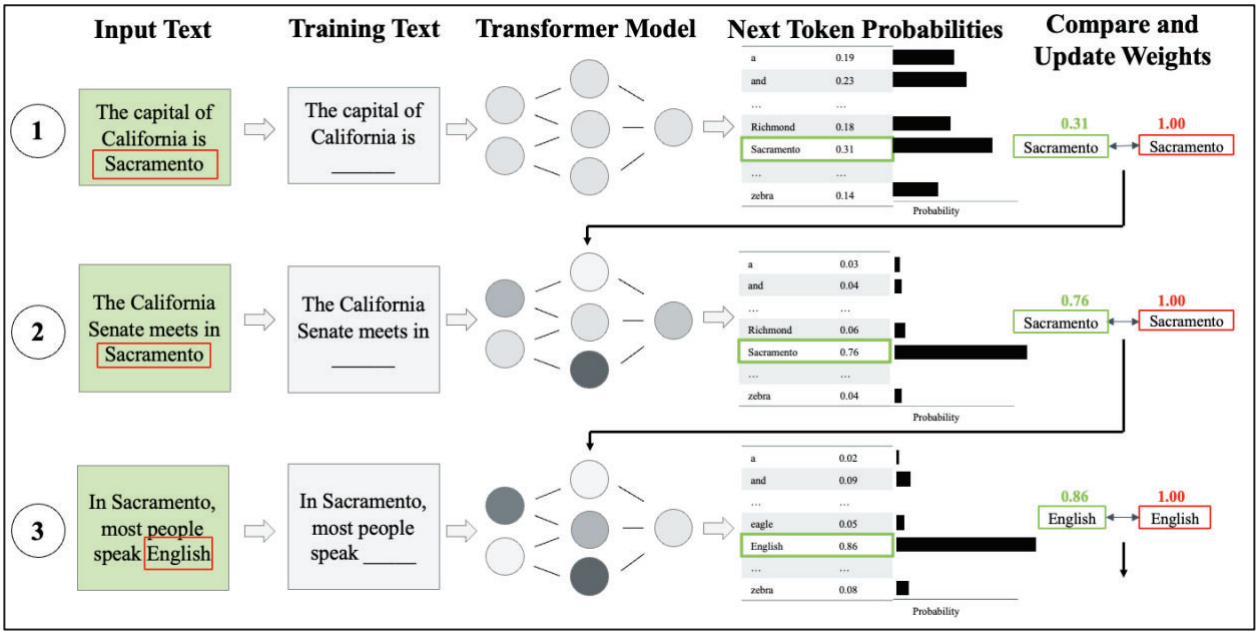
33. In this example, a complete text sequence, such as "We saw the Golden Gate Bridge on our trip to San Francisco," is obtained. Then, all of the tokens except the final one in the sequence ("We saw the Golden Gate Bridge on our trip to San," in this example) are input to the LLM, which produces probabilities for the next token to predict that was withheld from the input text sequence. In this simplistic example, the model's predicted probabilities result in the selection of "Jose" as the next token, which is then compared to the true final token of "Francisco". Based on this comparison, the model's weights are then adjusted given this incorrect assessment.[33]

34. However, this simplistic example fails to fully capture the information LLMs gain during training, as there are too many possible user prompts for LLMs to memorize every possible text input. As an example, consider merely the number of ways a user could ask an LLM what the

[31] Ungar Opening Report, ¶ 85.
[32] Ungar Opening Report, ¶ 86.
[33] Ungar Opening Report, ¶ 86.

capital of California is.[34] Additionally, while LLMs have billions of adjustable weights, those weights are insufficient to store responses to the infinite number of prompts users could input.[35] LLMs are instead designed to learn language patterns and knowledge that can be generalized to new input sequences.[36] The figure below shows an example of LLM generalization:

35. In this example, after being trained on the passage "The capital of California is



Sacramento" in Row 1, the model updates its weights to incorporate that information. Consequently, when prompted in Row 2 with "The California Senate meets in ____," the model is more likely to correctly respond with "Sacramento," as its weights were adjusted to better reflect the connection between California and Sacramento. The model is not simply memorizing that "Sacramento" completes the sentence "The capital of California is," but is building an internal association that leads to the probabilistic prediction that Sacramento is the capital of California. Those connections are leveraged to generalize to other situations.[37]

[34] "Where is the capital of California?" "Do you know what the capital of California is?" "What's the capital of California?" "What is California's capital?" "Where would I find the capital of California?" Ungar Opening Report, ¶ 87 n.125.

[35] Ungar Opening Report, ¶ 87.

[36] Ungar Opening Report, ¶ 88.

[37] Ungar Opening Report, ¶¶ 88, 90.

36. The same generalization idea is then applied in Row 3, in which the model has built an association leading it to predict that "English" completes the sequence, "In Sacramento, most people speak ____." It can then leverage this to generate appropriate continuations of prompts it never saw in training, generalizing to new contexts. That is, LLMs learn about language, grammar, and facts in training, allowing them to generalize their knowledge to new examples and contexts. Note, however, that this is an illustrative example—in practice, each individual token or passage of text adjusts the billions of weights of a model an infinitesimal amount, and it is only through training on many text examples that an LLM accumulates knowledge and associations.[38]

37. The LLM pretraining processes fundamentally transform the original training data into something completely different and unrecognizable from the original text. During training, input text is used to update the model's weights, which are not copies of the input text. Accordingly, when Meta distributes Llama models and weights to the open-source community, the training data is not included.[39] The training process is also extraordinarily computing-intensive, with Meta reporting that the training of Llama 3.1 required 38 septillion (38 followed by 24 zeros) computations. None of these computations would be required if the model was merely storing and regurgitating the input data. Other types of analysis and transformation of input data, such as creating indexes for categorizing, searching, and retrieving data, can be performed with a tiny fraction of the computing resources required to use that same input data to pretrain an LLM.[40]

38. I understand that under the copyright laws, "transformative uses" are uses that add something new to the original copyrighted work (that was allegedly infringed), with a further purpose or different character, and that do not serve as replacement for the original use of the copyrighted work. In my opinion, from a technical standpoint, Meta's use of copyrighted training data for research, evaluation, and training of Meta's LLMs provides a highly transformative use.[41] This is because, as discussed above, i) the text from the original copyrighted works is used in a new

[38] Ungar Opening Report, ¶ 88.
[39] Ungar Opening Report, ¶ 134.
[40] Ungar Opening Report, ¶ 140.
[41] Ungar Opening Report, ¶ 128.

way with an entirely different purpose, as the text is used to update Llama's weights when predicting the next token in a given sequence, and further is not stored in Llama, ii) the process of training Llama creates new meaning from the original text, as Llama learns statistical patterns and properties aggregated across the entire training data, and iii) the use of the text from the original copyrighted works does not replace its original use, as the data is used in a transformative way to create new text and applications, which does not serve as replacement for the original copyrighted material (as shown by the fact that Plaintiffs' works cannot be replicated using Llama, as I will explain in more detail below).[42] The downloading of these datasets in the first instance is also obviously a prerequisite to the use of the data for research, development, evaluation and training of Meta's LLMs.[43]

Overview of Meta's Llama Models

39. Beginning with Llama 1 in February 2023 and continuing through, most recently, Llama 3.3 in December 2024, Meta has released several versions of its Llama LLMs.[44] I have personal experience using and experimenting with these models, have reviewed source code for the models that was provided to me by Meta, and have read each of the papers published by Meta corresponding to them, in particular those papers published in conjunction with the release of Llama 1, Llama 2, and Llama 3. These materials confirm that the Llama models closely follow the general tenets of LLM processes and architecture that I described above.[45]

40. Llama 1 was released in sizes of 13, 33, and 65 billion parameters (or "13B, 33B, and 65B").[46] It was pretrained on a mix of publicly available datasets encompassing computer code, scientific papers, books, Wikipedia, and sections of "Common Crawl," miscellaneous content scraped from the Internet.[47] Llama 2 was released in July 2023 in sizes of 7 billion, 13 billion, and

[42] Ungar Opening Report, ¶ 129.
[43] Ungar Rebuttal Report, ¶ 86.
[44] Ungar Opening Report, ¶ 115.
[45] Ungar Opening Report, ¶ 113.
[46] Ungar Opening Report, ¶ 115.
[47] Ungar Opening Report, ¶ 118.

70 billion parameters.[48] Llama 2 shared similar architecture to Llama 1 and was pretrained on many of the same datasets as Llama 1, including Books3.

41. The first version of Llama 3 was released in April 2024 in sizes of 8B and 70B. Subsequently, Meta released Llama 3.1 in August 2024 in three sizes (8B, 70B, and 405B); Llama 3.2 was released in September 2024 in four sizes (1B, 3B, 11B, and 90B); and Llama 3.3 was released in one only size (70B) in December 2024.[49]

LLM Pretraining Data

42. While machine learning models generally require a substantial amount of data to inform accurate predictions, LLMs require a considerably larger training data set to ensure their responses are appropriate given their general purpose. This is driven by the wide variety of intended LLM use-cases: generating answers in response to questions from many domains, producing computer code, summarizing text, among many other uses. It is also driven by the scale of LLMs' neural networks, which contain billions of weights that must be finely tuned to generate appropriate responses. To do this effectively, these models must be trained on data that is sufficiently large, diverse, and high-quality. By the term "high-quality," I am not referring to data that human beings might subjectively find interesting, but rather to training data that will improve performance of LLMs.[50]

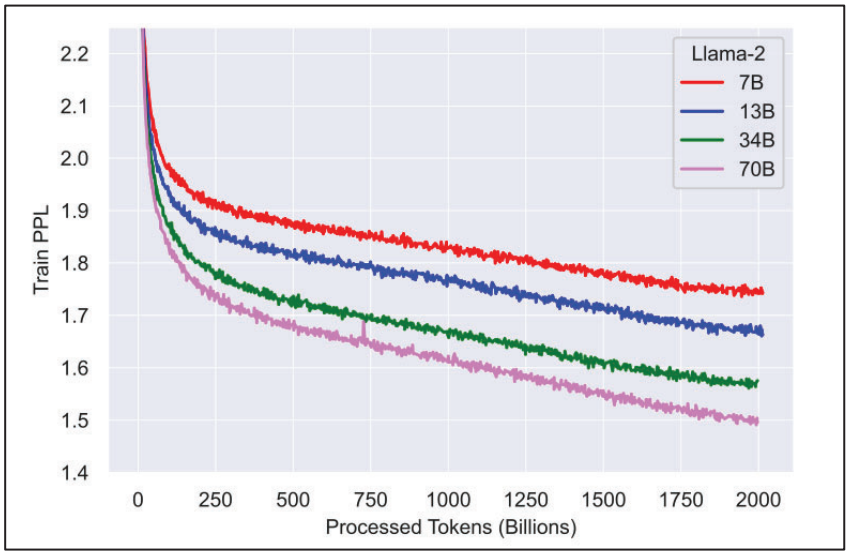
43. LLM training datasets are among the largest corpuses of text ever assembled. For example, Llama 2's collection of text (with 2 trillion training tokens) is approximately 50,000 times larger than the Encyclopedia Britannica and nearly 500 times larger than English Wikipedia. Llama 3 was pretrained on a significantly expanded, diverse corpus of more than 15 trillion tokens, which is 350,000 times larger than Encyclopedia Britannica and over 3,000 times larger than English Wikipedia. This amount of text is so large that if printed onto standard letter-sized paper, it would produce a stack of more than 930 miles high, which is approximately the distance between Los

[48] Ungar Opening Report, ¶ 113.
[49] Ungar Opening Report, ¶ 113.
[50] Ungar Opening Report, ¶ 90.

Angeles and Seattle. Moreover, training on this volume of data required immense computing power and energy. Most modern LLM datasets are of similar size, between 1 and 15 trillion tokens.[51]

44. The immense size of LLM training datasets is crucial because LLMs begin with random weights and must learn everything about language from scratch—word meanings, syntax, structure, and world knowledge—from statistical patterns in text alone. Achieving this requires a massive amount of data to capture the complexity and variety of human language, and to be able to effectively generalize what has been learned through training.[52] Increasing the scale of the training datasets and increasing the diversity of data a model sees in training improves its ability to generalize and its fluency with grammar and syntax.[53]

45. Moreover, scaling laws further show that increasing the training data size improves model performance across all LLM model sizes. This is illustrated by the figure below, which comes from the Llama 2 research paper released by Meta, illustrating that while larger models outperform smaller ones (based on having lower perplexity (PPL)), as the number of pretraining tokens increases, the models' performance continues to improve across all model sizes (7B, 13B, 34B, and 70B parameters):[54]



[51] Ungar Opening Report, ¶ 91.

[52] Ungar Opening Report, ¶ 91.

[53] Ungar Opening Report, ¶ 120.

[54] Ungar Opening Report, ¶ 91. I understand that a copy of the Llama 2 paper was attached as Exhibit G to the Declaration of Chaya Nayak, filed concurrently with this declaration. The figure shown in the text is from Figure 5 of the Llama 2 paper.

46. This figure shows that larger models outperform smaller ones, but also that training on additional data improves each model's performance. "PPL" on the y-axis is perplexity, where a lower perplexity value is better. None of the models here exhibited signs of saturation (*i.e.*, rising or flattening perplexity), indicating that further training on additional data would further enhance their performance, and demonstrating the need for massive datasets to create high performing LLMs that perform well on diverse tasks.[55]

47. LLM training data must also be extremely diverse, meaning it includes texts from many different domains, styles, languages, and contexts. An LLM trained on only one type of data would not perform well on general purpose tasks. If a model only saw social media posts, for example, it would not do well in generating source computer code.[56]

48. I understand that a small portion of the training data for Llama was comprised of books, some of which were sourced from a dataset known as "Books3," a commonly used LLM training dataset that contains tens of thousands of books. Books3 is one of the 22 subsets of The Pile dataset, a large, diverse text dataset of nearly 1 trillion words assembled by EleutherAI, a non-profit AI research group, and composed of text from various sources such as encyclopedias, webpages, social media, and movie subtitles, among others. Today, over 100 organizations, including technology companies, universities, research institutions, and non-profits have published AI-related research utilizing the Pile and Books3, which collectively have been cited over 10,000 times.[57]

Llama's "Memorization" Rates are Extremely Low

49. In the context of LLM, the term "memorization" is sometimes used to refer to the ability of an LLM to produce output that matches its training data. Memorization in this context occurs when LLMs overfit on the average probabilities of specific training data.[58] LLMs tend to

[55] Ungar Opening Report, ¶ 91.

[56] Ungar Opening Report, ¶ 93.

[57] Ungar Opening Report, ¶ 236.

[58] Ungar Opening Report, ¶ 199. "Overfitting" occurs when models learn their training data too well, incorporating statistical noise and minor fluctuations in the training data. Such overfitting leads to the model performing well on training data but poorly on unseen testing data, indicating poor generalization. Ungar Opening Report, ¶ 58.

memorize facts, phrases, and texts that appear very frequently in their training data and are thus frequently useful for next token prediction. For example, Llama is trained on many texts that contain quotations from the U.S. Constitution. Because text from the Constitution is found thousands of times in its training data, Llama's weights are slowly adjusted during training such that it usually correctly predicts the next token in sequences from the Constitution. The same is true of other very well-known and frequently repeated texts, such as the Gettysburg Address, the beginnings of famous books (*Moby Dick*, *Frankenstein*, and others), well-known poems (*The Road Not Taken,* for example), and other frequently repeated texts that appear in multiple places on the Internet and thus are repeated multiple times in the training datasets.[59]

50. But the term "memorization" is a misnomer, as LLMs, and Llama specifically, do not function by memorizing training data. Rather, Llama learns statistical patterns and relationships from the training data, generating predictions by assigning probabilities to possible next tokens based on the provided context, and using these probabilities to predict the next token. Unlike a fixed memory lookup, LLMs rely on a sampling strategy to produce next token predictions, meaning there is no single deterministic sequence of tokens attributed to a model.[60]

51. Researchers have extensively studied LLMs' memorization abilities and behavior to quantitatively and qualitatively describe the content LLMs memorize. Many measure ***discoverable memorization***, which checks whether the LLM completes a text verbatim when prompted with the beginning of the text. Under discoverable memorization, texts are considered memorized if, when provided the first 50-100 tokens in a text, the LLM accurately reproduces the next 50 tokens from that text. Discoverable memorization is favored because it is an efficient method of testing whether a specific text is memorized, which allows researchers to study memorization behavior by type and frequency of text. To test an LLM's rate of discoverable memorization researchers sample passages from the LLM's training data and prompt the LLM with the beginnings of those passages. The LLM's outputs are then compared with the true continuation of the text; if the model accurately continues the text for a predetermined number of tokens (often

[59] Ungar Opening Report, ¶ 199.

[60] Ungar Opening Report, ¶ 198.

50), the text is considered memorized.[61] I also observe that Plaintiffs' expert, at her deposition, described this technique as a "commonly accepted methodology for studying the phenomenon of memorization in LLMs." (Lopes Depo., 02/13/2025, 105:6-12.)

52. I conducted experiments to assess whether books authored by Plaintiffs are "memorized" by Llama models,[62,63] the most expansive of which is described here. All other experiments demonstrated similar conclusions.[64] As detailed in my rebuttal expert report, I selected a work from each of the 13 Plaintiffs and, for each work, evaluated memorization across all text contained in those works. Each passage was input into the model and the model's output was compared with the correct completion. For each of these 13 books, the first 150 tokens (tokens 1-150) in the book were input to Llama, and Llama's 50-token output is compared with the next 50 tokens (tokens 151-200) from the book, to compute the continuation length. Then, the input passage was shifted one token down (to tokens 2-151) and again input to Llama, with its output being compared with the next 50 tokens (tokens 152-201) to compute the continuation length.[65] The model is scored by the number of tokens it outputs that match the actual continuation sequentially. For example, if the passage "The quick brown fox" was input to the model, and the correct continuation was "jumped over the lazy dog," and the model's output was "jumped over the *tired* dog," the model's continuation length score would be 3. Then the passage was shifted down another token down (to tokens 3-152), input to Llama, and so on, continuing through the entire book. This experiment used the Llama 3 70B model,[66] because it scored the highest, on average, on the earlier continuation experiments described in my opening report and thus provides an approximate upper bound on the continuation performances of models used in the earlier

[61] Ungar Opening Report, ¶ 200.

[62] Ungar Opening Report, Section V.D.3.

[63] Ungar Rebuttal Report, Section III.A.2.d.

[64] Ungar Opening Report, ¶ 195.

[65] Ungar Rebuttal Report, ¶ 22.

[66] This experiment is extremely computationally intensive, since it analyzes one prompt and response for each token in each of the selected works by plaintiffs to this case. The selected books contain between 31,496 and 157,128 tokens each, and the full experiment produced over 1.2 million total prompts and responses.

experiment[67] All of these tests were conducted using Llama base models, *i.e.*, LLMs that have only undergone pretraining with the next-token prediction objective, without any finetuning for chatting or instruction-following. Base models continue passages by default.[68]

53. The table below shows average continuation length (in tokens) using Llama 3 70B for both experiments; as shown, Llama on average was only able to complete a single token:[69]

Plaintiff's Work		Llama 3 70B Average Continuation Length
Author	Title of Work	
Richard Kadrey	*Sandman Slim*	0.88
Sarah Silverman	*The Bedwetter*	0.97
Christopher Golden	*Ararat*	0.87
Ta-Nehisi Coates	*The Beautiful Struggle*	0.62
Junot Diaz	*Drown*	0.79
Andrew Sean Greer	*The Confessions of Max Tivoli*	0.60
David Henry Hwang	*M. Butterfly*	1.56
Matthew Klam	*Who is Rich?*	0.64
Laura Lippman	*After I'm Gone*	0.77
Rachel Louise Snyder	*No Visible Bruises: What we don't know about domestic violence can kill us*	0.94
Jacqueline Woodson	*Brown Girl Dreaming*	0.87
Lysa TerKeurst	*Embraced*	2.52
Christopher Farnsworth	*Blood Oath*	0.86
Overall	**Average**	**0.99**

54. The table below, focusing on the second (expanded) experiment that covers all of the text of these works, shows the number of passages in each work continued for 50 tokens or greater, along with the total number of tokens included in these passages:[70]

[67] Ungar Rebuttal Report, ¶ 24.

[68] Ungar Rebuttal Report, ¶ 23.

[69] Ungar Rebuttal Report, Table 6, Appendix C.

[70] Ungar Rebuttal Report, ¶ 26, Table 1. Because this second experiment tested every passage in the selected books, incrementing one token at a time, some passages overlap, so I consolidated the results so that overlapping passages were counted only once. The continuations ranged in length from 60 tokens to 148 tokens, with an average length of 70 tokens. Ungar Rebuttal Report, ¶ 25.

Author	Title of Work	Number of Tokens in Work	Number of Passages Outputted	Number of Tokens Outputted	Outputted Tokens as % of Total Tokens
Richard Kadrey	*Sandman Slim*	126,102	0	0	0.0%
Sarah Silverman	*The Bedwetter*	72,790	1	148	0.2%
Christopher Golden	*Ararat*	116,694	0	0	0.0%
Ta-Nehisi Coates	*The Beautiful Struggle*	64,854	1	85	0.1%
Junot Diaz	*Drown*	55,446	1	77	0.1%
Andrew Sean Greer	*The Confessions of Max Tivoli*	110,678	0	0	0.0%
David Henry Hwang	*M. Butterfly*	31,446	2	168	0.5%
Matthew Klam	*Who is Rich?*	117,033	0	0	0.0%
Laura Lippman	*After I'm Gone*	123,798	0	0	0.0%
Rachel Louise Snyder	*No Visible Bruises: What we don't know about domestic violence can kill us*	157,078	2	111	0.1%
Jacqueline Woodson	*Brown Girl Dreaming*	40,406	1	60	0.1%
Lysa TerKeurst	*Embraced*	91,414	30	2,017	2.2%
Christopher Farnsworth	*Blood Oath*	127,766	0	0	0.0%
Overall	Average (including *Embraced*)	95,039	2.9	205	0.3%
	Average (not including *Embraced*)	**95,341**	**0.7**	**54**	**0.1%**

Embraced, by Lysa TerKeurst, is an outlier in these results. This is attributable to the large number of quotes from The Bible that appear in that work – of the 30 completed passages reported above, 25 are excerpts from the Bible. As The Bible is frequently repeated in Llama's pretraining data, Llama is thus better able to continue these passages.[71]

55. I also note that the design of my continuation experiments is an unlikely and adversarial use case for LLMs. This is because it involves prompting an LLM to continue a particular passage from a book; the majority of the passage is actually provided to the model in the prompt entered by the user, which means the underlying book is *already* in the possession of that

[71] Ungar Rebuttal Report, ¶ 25; Ungar Opening Report, ¶ 158 n.246.

user. This type of prompt is a standard way to scientifically test the ability of an LLM to output a particular work, but it otherwise serves no practical use case or purpose.

56. Rather, such a prompt is an example of an "adversarial" prompt, which refers to a prompt designed to manipulate an LLM into producing an untended result—in this case outputting the text of a particular work on which it was trained. While continuation tests are not reflective of typical LLM usage, they provide a means to assess whether an LLM is able to output specific text. Thus the continuation results presented here overstate the model's propensity to output text from Plaintiffs' works, as compared to normal usage by users.[72]

57. Evidence from Meta's internal research and third-party studies suggests that passages Llama continued for 50 or more tokens are likely to be repeated frequently in Llama's training dataset. To test this hypothesis, each identified passage that Llama was able to complete was searched for on the internet, and the number of individual websites storing copies of the passage were counted. For example, the passage from *The Bedwetter* that is continued for 50 tokens is found on several websites, including goodreads.com,[73] thenewinquiry.com,[74] and academia.com.[75]. The same is true for every passage in all the selected works by Plaintiffs to this case which Llama 3 70B continued for 50 tokens or greater. Each was found on multiple openly available websites and thus could be easily read in these third-party sources. The table below lists the number of passages that were continued for 50 tokens or greater, by author and work, and the availability of these in the public domain.[76]

[72] Ungar Opening Report, ¶ 178; Ungar Rebuttal Report, ¶ 22.

[73] "A Quote from The Bedwetter," Goodreads, accessed January 29, 2025, https://www.goodreads.com/quotes/529142-at-some-point-i-figured-that-it-would-be-more.

[74] Esme Douglas, "Laughing at America," The New Inquiry (blog), January 31, 2018, accessed January 29, 2025, https://thenewinquiry.com/laughing-at-america/.

[75] Megan Phipps, "Kurt Vonnegut's Slaughterhouse Five as Postmodern Historiographic Metafiction," January 1, 2014, https://www.academia.edu/31994517/Kurt_Vonneguts_Slaughterhouse_Five_as_Postmodern_Historiographic_Metafiction.

[76] Ungar Opening Report, ¶ 28.

Author	Title of Work	Number of Pages in Google Books Preview	Number of Passages Outputted	% of Outputted Passages found on Web	% of Outputted Passages in Google Books Previews[77]
Richard Kadrey	*Sandman Slim*	10	0	n/a	n/a
Sarah Silverman	*The Bedwetter*	20	1	100%	0%
Christopher Golden	*Ararat*	32	0	n/a	n/a
Ta-Nehisi Coates	*The Beautiful Struggle*	24	1	100%	100%
Junot Diaz	*Drown*	22	1	100%	100%
Andrew Sean Greer	*The Confessions of Max Tivoli*	29	0	n/a	n/a
David Henry Hwang	*M. Butterfly*	15	2	100%	100%
Matthew Klam	*Who is Rich?*	33	0	n/a	n/a
Laura Lippman	*After I'm Gone*	10	0	n/a	n/a
Rachel Louise Snyder	*No Visible Bruises: What we don't know about domestic violence can kill us*	54	2	50%[78]	50%
Jacqueline Woodson	*Brown Girl Dreaming*	32	1	100%	100%
Lysa TerKeurst	*Embraced*	38	30	100%	96.67%
Christopher Farnsworth	*Blood Oath*	33	0	n/a	n/a

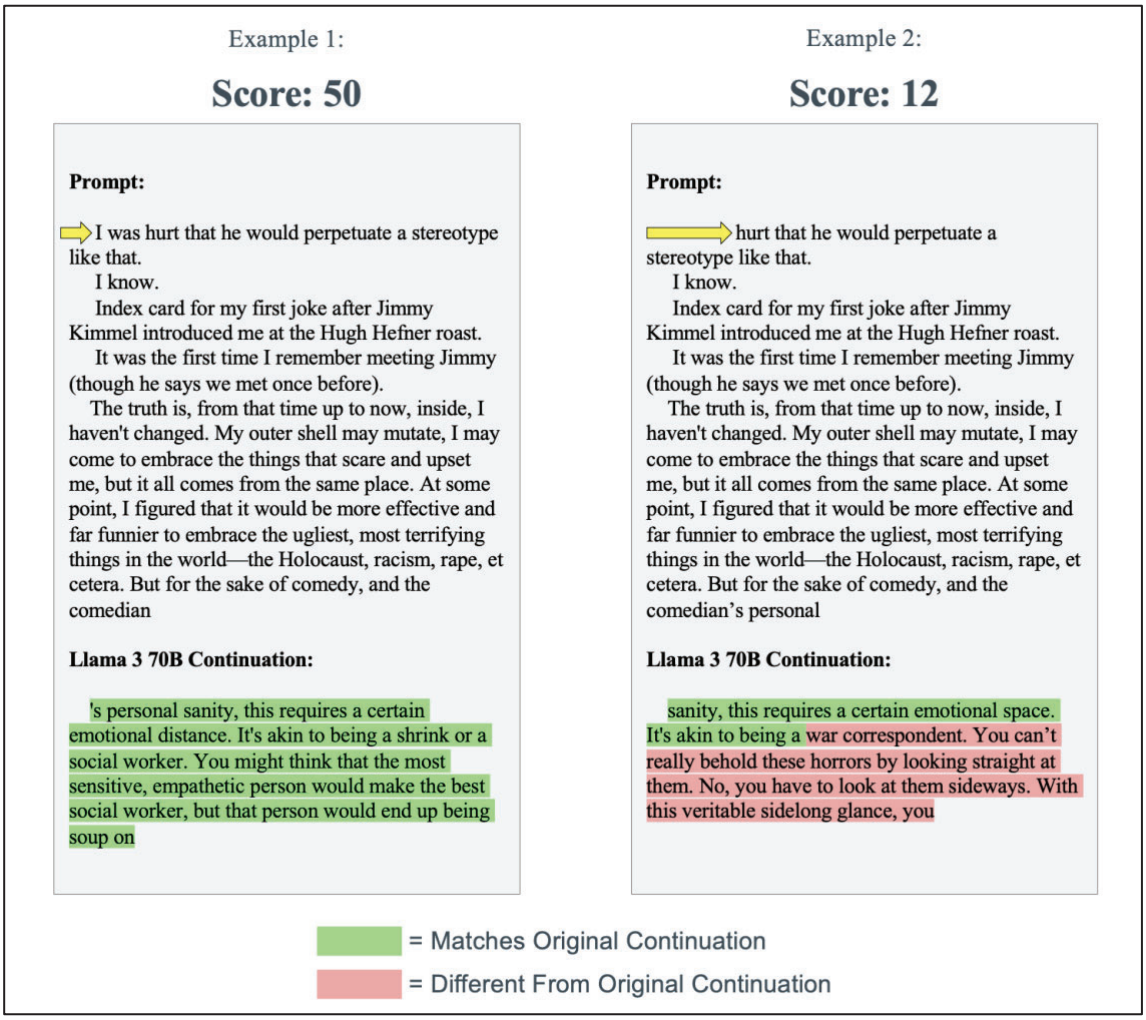
58. In addition to demonstrating that a very small percentage of text from Plaintiffs' works can be continued for 50 tokens or greater, the results from my expanded continuation test also highlight the variability present in the model's ability to continue passages, which in turn demonstrates the implausibility of using Llama to read or obtain works like the Plaintiffs' books.[79]

[77] Some outputted passages appear exactly in Google Books previews of works that are not authored by the Plaintiffs. As I discuss above, passages with continuation lengths of 50 or greater are likely to be repeated frequently in Llama's training dataset, and certain of these Google Books Previews, although not authored by the Plaintiffs, serve as examples of how such repetitions occur across the internet and may be present in multiple sources. Ungar Rebuttal Report, ¶ 29 n.59.

[78] One of the outputted passages for *No Visible Bruises* is the word "here" repeated 50 times. Further, the 150 token prompt used for this section of Rachel Louise Snyder's work includes text that is highly repetitive. Therefore, the model outputting the text correctly is not necessarily an indication that the model is familiar with the specific work, but rather that the model was able to output the simple pattern repeated in the prompt. Ungar Rebuttal Report, ¶ 29 n.60.

[79] Ungar Opening Report, ¶ 27.

59. One example from *The Bedwetter* by Sarah Silverman is shown below, in which shifting the input passage by only two tokens reduced the model's continuation score from 50 to 12, meaning rather than continuing the passage for 50 tokens (approximately 50 words), the model instead continues the passage for only 12 tokens (about 12 words).[80]



The two excerpts from *The Bedwetter* are shown on the left and right, with the input prompt (passage) offset by only two tokens. Llama 3 70B was induced to continue the passage for 50 additional tokens, with its continuations shown underneath the excerpts. Continuation text is highlighted green if it matches the original continuation and red if it differs. The example on the left is successfully continued for 50 tokens, but the one on the right, which is offset by only two tokens from the first example, is continued for only 12 tokens, demonstrating the instability seen

[80] Ungar Opening Report, ¶ 27.

in continuation tests.[81] As stated above, this demonstrates the implausibility of using Llama to read or obtain works like the Plaintiffs' books.

Individual Works Have No Measurable Impact on the Llama Training Process

60. I understand that the Plaintiffs in this case and their expert have asserted that the Llama models exist as they are, partly because of the expressive choices in Plaintiffs' works, but this statement is unsupported. Individual books typically have between 50,000 and 150,000 tokens,[82] and thus represent a tiny portion of the trillions of tokens used to train the Llama models. LLMs are trained across billions of steps via a complex, computation-intensive process to predict the next token in a sequence. Each of the billions of neural network weights that contribute to the LLM's prediction is updated at each training step, including the model's word embeddings which begin as random numbers and are gradually adjusted throughout training. The training step is repeated billions, even trillions of times (based on the size of the training dataset) until the LLM has made predictions about all the trillions of words in its training dataset.[83] LLMs are simultaneously trained on many text examples, called batches. For example, Llama 2 was trained on batches of 4 million tokens. This broader exposure to more training data helps stabilize the model's optimization process, by allowing the model to make generalized adjustments, rather than specifically adjusting its weights to adapt to a single, specific training example. Because sequences from the Plaintiffs' works are used in training as part of batches of over a billion tokens, combined with thousands of other texts from many other sources, it is not possible to determine if any text sequence from the Plaintiffs' works had an effect on Llama's weights.[84] For the same reasons, it is highly unlikely that any individual work in the training corpus had a measurable effect on model performance.

61. I was also asked to conduct an experiment to address the question of the likely impact, if any, of books like the Plaintiffs' books on the performance of the Llama models. To do

[81] Ungar Opening Report, ¶ 27.

[82] Ungar Opening Report, ¶ 169.

[83] Ungar Rebuttal Report, ¶ 33; Ungar Opening Report, ¶ 85.

[84] Ungar Rebuttal Report, ¶¶ 36-37.

so, I designed an experiment that uses a technique known as "continued pretraining," which continues to apply pretraining to a released base model. Continued pretraining is frequently used by practitioners to supplement the model with additional, specific pretraining data to a model. In my experiment, I continued the pretraining of Llama models with works comparable to those of the Plaintiffs, while preserving knowledge the model acquired during pretraining.[85] These additional works were selected based on having similar length, genre, and approximate popularity to Plaintiffs' books. I then performed evaluations conducted to gauge the models' performance before and after continued pretraining using the industry-standard benchmarks known as MMLU (Massive Multi-Task Language Understanding). I used the MMLU assessment because of its general-purpose nature and its ubiquitous usage within the machine learning community, as it has over 10,000 questions across 57 diverse domains. Further, MMLU is one of the only benchmarks with published results by Meta in all of its Llama papers, and so using MMLU allows for a consistent baseline evaluation across my analysis results. Results from my experiments demonstrated that, after continued pretraining on new books, adding a new book comparable to Plaintiffs' works to Llama's training set adjusted its performance by less than 0.06% on industry standard benchmarks, a meaningless change no different from noise.[86]

62. I also empirically assessed the impact of pretraining on a particular work on Llama's ability to continue passages of text directly from that work. To quantify this, I measured the model's ability to continue excerpts from the training works that were used for continued pretraining—in this case, on fiction books. A continuation test, as described previously, quantifies

[85] The reason I chose continued pretraining for other books that were comparable to Plaintiffs' books is because the Plaintiffs have alleged that the Llama models were already trained using their books. I considered LLM "unlearning" techniques currently being investigated by researchers but found them generally unsuitable here. Performing experiments using continued pretraining on books comparable to Plaintiffs' books nevertheless provides a reliable indicia of the impact of using Plaintiffs' books for training (and more generally, the impact of using any individual book) on the performance of the Llama LLMs. Ungar Opening Report, ¶ 149 n.235.

[86] To contextualize this result, trials were also conducted in which the Llama model was trained on random, useless text (such as repeated strings of one character or word, e.g., "ttttt", "the the the", etc.), instead of additional books, and MMLU evaluations after these tests show a range of noise results which fully encapsulate the MMLU variance from tests on the chosen books. This comparison indicates that benchmark variations after continued pretraining on a single work are no different than noise. Ungar Opening Report, ¶ 155.

the extent to which an LLM can produce passages from a work. It is conducted by taking excerpts from the work and prompting the model to continue them with the correct continuation as it appears in the original work. The model is scored by the number of tokens it outputs that match the actual continuation sequentially. For example, if the passage "The quick brown fox" was input to the model, and the correct continuation was "jumped over the lazy dog," and the model's output was "jumped over the *tired* dog," the model's score would be 3. Prior to presenting the selected passage, an instruction is given to the model to continue the excerpt.[87]

63. To assess the model's overall ability to continue passages from a work, I sampled 100 passages from random points within the books, each 150 tokens long, and presented each excerpt individually to the model. Each passage produced a continuation score, and I averaged all 100 scores to create a final measure for each book. For example, if a model scored 5.8 on *The Great Gatsby*, that means that, on average, it accurately continues passages from *The Great Gatsby* for 5.8 tokens over 100 trials when prompted with a 150-token passage before its output diverges from the correct continuation. Recall that a token is approximately one word, so the phrase "We went to the park." is six tokens (the period is a separate token).[88]

64. The results of this experiment were that before continued pretraining on the selected works, when provided with a 150-token passage randomly sampled from the work and asked to continue it, the Llama models that I tested produced fewer than two tokens (on average, over 100 samples) before diverging from the true continuation, across the models that I tested. To contextualize what one and two tokens are, if we consider the start of a sentence to be "We went" a one token continuation would be the word "to", and a two token continuation would be "to the". After adding the selected books to Llama's training data via continued pretraining, Llama was able to produce 0.01 tokens fewer (that is, it was less able to reproduce text from the selected works after continued pretraining on those exact works).[89]

[87] Ungar Opening Report, ¶¶ 175-176.

[88] Ungar Opening Report, ¶ 177.

[89] Ungar Opening Report, ¶¶ 187-189.

Meta's Removal of High-Frequency Text From Training Data is Common Practice

65. LLM training datasets like those used by Meta to train Llama contain trillions of words. Before training, practitioners typically deduplicate these datasets, meaning they remove all but one copy of texts from the dataset. Deduplicating datasets has been shown to improve LLMs' performance on real-world tasks, but the process of deduplicating across trillions of words is complex and time-intensive.

66. High-frequency texts are lines and paragraphs of text that, without filtering, would otherwise occur thousands or millions of times in a dataset. Removing high-frequency texts before deduplication reduces the overall size of the dataset in the computationally efficient way, making future deduplication efforts more computationally efficient (because the number of documents to be deduplicated is smaller). Examples of high-frequency texts include open-source software licenses, boilerplate terms and conditions statements, copyright notices, and website privacy notices, among many others.[90]

67. As part of their deduplication pipeline, practitioners often attempt to remove all instances of these high-frequency texts by programmatically removing lines and paragraphs that include keywords suggesting they contain high-frequency text. For example, while processing its FineWeb dataset, Hugging Face engineers removed all paragraphs that contained "terms of use," "privacy policy," and "cookies policy," likely because those paragraphs were very likely to contain boilerplate, frequently repeated text.[91]

68. Importantly, high-frequency texts are not removed because practitioners do not want their LLM to be exposed to them at all. Continuing the example above, it is important for an LLM trained on web data to see privacy policies and terms of use statements in training. However, without programmatically filtering paragraphs with those terms, those passages would be repeated thousands of times in the training data, which would cause the model to overfit to those passages.[92]

[90] Ungar Opening Report, ¶ 218.
[91] Ungar Opening Report, ¶ 218.
[92] Ungar Opening Report, ¶ 219.

69. Meta implemented a variety of techniques to remove various types of duplicated works and high-frequency texts. Such text also includes lines that contain excessive repetition (less than 8% unique words) and excessive newline characters. To the extent that copyright-related information was removed from training data by Meta, I have seen no indication that it was anything other than a broader data pre-processing and deduplication effort to remove repetitive data from a training dataset.[93]

Meta's Open-Source Release of Llama Benefits Innovation and Research

70. LLMs are classified as either closed-source or open-source. The distinction between closed-source and open-source is a longstanding dichotomy in computer science that predates LLMs by decades. Using the example of computer operating systems, Microsoft Windows is generally regarded as closed-source because Microsoft does not make the source code of the operating system available to the public. Linux, on the other hand, was released as an open-source operating system in the 1990s, and today, is freely available through the open-source community. This means anyone can download source code for Linux, and then study it, modify it, and redistribute it, subject to the governing open-source license. The open-source nature of Linux allows users and developers to experiment with and customize the operating system in ways that would be infeasible for closed-source alternatives such as Microsoft Windows. Software distributed through an open-source model is typically governed by an open-source license (which usually accompanies the software files themselves), and the license specifies the conditions applicable to the use and/or redistribution of the software.[94]

71. In the context of LLMs, ***closed-source models*** are accessible only through provided access points, such as web interfaces and through Application Programming Interfaces (APIs), which allow third-party applications to access a closed-source LLM. These APIs, for example, allow third-party applications to transmit prompts to a closed-source LLM (for example by transmitting them over the Internet), to which the LLM responds and provides output (for example, by sending the response to the third-party application over the Internet). But a third-party developer

[93] Ungar Opening Report, ¶ 220.

[94] Ungar Opening Report, ¶ 108.

cannot, for example, download the LLM and execute it on their own computer systems. As a result, much of the internal operation of the LLM is a "black box" from the perspective of the third-party application developer or user. This allows a provider of a closed-source LLM to maintain control over access to the LLM and, thus, also allows the provider to charge a fee for accessing and using the LLM if it so desires. Because the exact architectures and trained weights of closed-source LLMs are not made public, developers have limited ability to modify the behavior of the LLM or adapt them to their own context or data, and experimenting on these types of LLMs can be cumbersome and limiting. Examples of closed-source LLMs include OpenAI's ChatGPT models, Anthropic's Claude models, and Google's Gemini models.[95]

72. In contrast, ***open-source models*** are publicly available: open-source LLM providers explicitly share their exact LLM architectures and distribute their LLM's weights, generally for the benefit of the scientific community and research. Within the terms of a provider's license, open-source LLMs can be downloaded and actually run on a developer's own computer systems, allowing the developer to study, experiment, modify and adapt them to new contexts. Llama is an example of an open-source LLM, and its open-source release has provided an enormous benefit to the scientific community and spawned countless other AI research projects.

73. An important aspect of Llama's popularity and usage is the fact that it is open-source. Llama open-source models include modules necessary for development and research on the models, including machine learning model code, trained model weights, inference-enabling code, training-enabling code, fine-tuning enabling code, and other elements that comprise the LLM. LLMs are extremely expensive to train, beyond the resources of most academics and individuals. However, many research applications of LLMs require direct access to the model's weights.[96]

74. Open-sourcing Llama models brought a range of benefits across research and innovation, industry adoption, and transparency and trust. I observed in my opening report that the papers behind Llama 1, Llama 2, and Llama 3 were cited on Google scholar 11,800 times, 11,200

[95] Ungar Opening Report, ¶ 109.

[96] Ungar Opening Report, ¶ 110.

times, 1,700 times respectively.[97] Moreover, the Llama 1 and Llama 2 models have individually been cited by more than 7,000 other research publications, indicating the significance of free access to the models and the ability by practitioners to utilize them for new research.[98]

75. A significant advantage of open sourcing Llama is that the project can be used to develop research by the broader technology community—practitioners and engineers from diverse backgrounds, beyond just at Meta, can contribute to the model's development, find and fix bugs, and adapt it for specialized use cases. This collective ability to shape and leverage such foundational technology brings greater efficiencies in furthering AI and its impact across a range of industries.[99]

76. Open access to these modules contributes to AI research and innovation in a number of ways, including by enabling the reproducibility of scientific results. For example, using the Llama 2 model, practitioners from UC Berkeley were able to develop a new and more effective specification, referred to as "Retrieval Augmented Fine Tuning" (RAFT) for fine-tuning LLMs to specialize in a single domain. In part because the underlying research was reproducible, RAFT has been highlighted as a superior methodology over previous methods.[100]

77. Open sourcing Llama also enables greater experimentation and industry adoption. Industries like education, law, finance, healthcare, and technology, among others, have leveraged Llama to quickly experiment, prototype, and deploy Llama-based applications at great scale. For example, "PMC-LLaMa" provides state- of-the-art performance on evaluations related to healthcare (questions and answers from the U.S. Medical License Exams) and is developed by introducing healthcare specific knowledge to Meta's Llama models. Additionally, practitioners developed "ChatDoctor" an LLM trained to answer queries from patients based on their symptoms, utilizing the open-source Llama 2 models. These models have been adopted as baselines for other research aimed at developing LLMs for applications in the healthcare industry. As a result, practitioners involved in the creation of these models not only paved the way for applications of

[97] Ungar Opening Report, ¶ 245.
[98] Ungar Opening Report, ¶ 248.
[99] Ungar Opening Report, ¶ 247.
[100] Ungar Opening Report, ¶¶ 249-251.

LLMs in the healthcare industry, but also developed finetuning methodologies rich with medical data for training similar models in the future.[101]

78. One of the most impactful aspects of open sourcing Llama is that it provides democratized access to individuals, startups, and smaller companies that could not afford closed source models, or which require customization to specific needs. Such benefits enable greater innovation more quickly, without having to develop the infrastructure required for LLMs for every application, which would be cost-prohibitive in most cases.[102]

79. For example, in healthcare, organizations from developing nations have adopted the Llama models to provide guidance around maternal health for people in remote areas. Organizations like Jacaranada Health, utilized the open-source Llama 2 model to train "UlizaLlama," finetuned on prior data containing their interactions with African mothers in Swahili. The resulting model understands Swahili and is able to respond with questions on maternal health with increased accuracy and sensitivity.[103]

80. Finally, and importantly, open sourcing Llama enables greater transparency and trust in the development of LLMs. Open sourcing the models has allowed researchers to better understand its innerworkings and create solutions to critical questions related to LLM safety, fairness, and compliance. For example, while mechanisms exist to control for harmful use cases of LLMs, malicious actors have discovered ways to bypass these mechanisms. Researchers with access to open-source models, then, can reciprocate the environment for the LLM to generate harmful outputs and study their behavior to develop strategies for the LLMs to be resistant to similar attacks in the future.[104]

81. Attached hereto as Exhibit A is a true and correct copy of my current curriculum vitae.

[101] Ungar Opening Report, ¶¶ 253-255.
[102] Ungar Opening Report, ¶ 256.
[103] Ungar Opening Report, ¶ 257.
[104] Ungar Opening Report, ¶ 260.

I declare under penalty of perjury that the foregoing is true and correct. Executed on this 24th day of March, in Philadelphia, Pennsylvania.

/s/ Lyle Ungar, PhD